DAS ELEKTRONISCHE MAGAZIN 11/85

INPUT 64

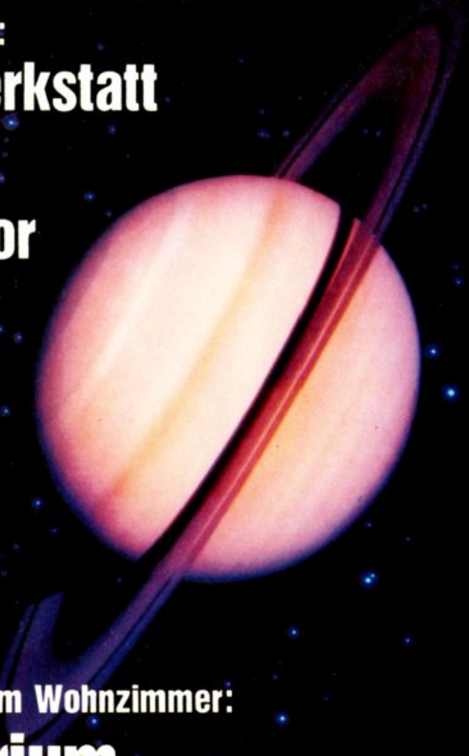
Infos · News · Programme · Unterhaltung · Tips

Hilfsprogramm:
BASIC-Zeilen-Generator

Serien
Spiele
Kurse

Neu in INPUT 64:
Die ID-Werkstatt

Disketten retten:
Disc-Doctor



Sternenhimmel im Wohnzimmer:
Planetarium

Dokumentation und Bedienungshinweise

INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61

INHALTSVERZEICHNIS INPUT 64

11/85

Liebe Leser!

Sie haben einen Commodore C64? Wir auch! Sie haben ein Floppy-Laufwerk 1541? Wir auch! Und natürlich eine Datasette von Commodore. Haben wir auch! Gemeinsame Freud, gemeinsames Leid: Wo Gleiches draufsteht, ist nicht unbedingt Gleiches drin.

Charakterliche Abweichungen gehen wie immer vom Innenleben aus. Die Konstrukteure von Betriebssytemen entwickeln sich eben weiter, was zur Folge hat, daß vom C64 mindestens zwei verschiedene Betriebssysteme auf dem Markt sind, die sich in kleinen schwarzen, schwer durchschaubaren Chips, den Kernal-ROMs verstecken. Wer Mut genug hat, seinem Rechner die Schrauben herauszudrehen, die das Gehäuse zusammenhalten, findet oben links drei kleinere 'Käferchen' mit 24 Beinen. Im Mittleren schlummert das Kernal, flankiert zur Rechten vom BASIC-ROM und zur Linken vom Zeichensatz. Die mittlere Kennziffer, die auf dem Chip aufgedruckt ist, offenbart Ihnen durch die Endung 02 oder 03, welche Version des Betriebssystems für die Arbeitsweise Ihres Rechners verantwortlich ist. Die Version 01 soll, nach Auskunft des Herstellers, den deutschen Markt nie erreicht haben.

Wer die Werkzeugkiste lieber im Schrank lassen möchte, kann durch POKE 1024,0 nach der Einschalt-Meldung die Version seines Betriebssystems entlarven. Version 03 reagiert mit einem '∫' in der oberen linken Ecke als erstes Zeichen auf dem Bildschirm. Bei der Version 02 ist die Wirkung scheinbar gleich Null. Erst wenn Sie den Cursor mit HOME auf die Stelle positionieren, wird das erwartete Zeichen sichtbar. Der Grund: Werden Bildschirm-Zeichen direkt in den Speicher 'gepoked', nimmt die Version 03 die aktuelle Zeichenfarbe, während Version 02 die Zeichenfarbe des Hintergrunds verwendet. Ein blaues Zeichen auf blauem Hintergrund ist nun mal für das Auge unsichtbar.

Obwohl das elektronische Innenleben der Datasette nicht sehr kompliziert ist, gibt es auch hier bei den Geräten des Hauses Commodore Unterschiede. Das Gerät mit kleinerem Bandzähler reagiert empfindlicher auf die Aufzeichnungsart der Daten auf dem Band. Die kleine elektronische Datenfabrik - der Schmitt-Trigger - ist bei neueren Geräten toleranter.

Die Platinen der Floppy-Laufwerke, die einen eigenen Prozessor (6502, der ältere Bruder vom 6510 im C64) und eigene ROMs tragen, sind in letzter Zeit beliebter Tummelplatz für Experten geworden. Auch hier gab es verschiedene ROM-Versionen, doch entscheidend sind die Eigenarten des Prozessors, der für die 'Intelligenz' der Floppy zuständig ist. Es gibt eine Vielzahl von Herstellern und Ausgaben dieses Prozessors, der übrigens in vielen Home-Computern seine Arbeit verrichtet. Einige dieser Prozessoren verstehen auch ungewöhnlichere Befehle, die normalerweise nicht zu Ihrem Wortschatz gehören, während andere auf solche Ansprüche mit schlichter Untätigkeit reagieren. Besonders trickreiche Programme überstehen so den Übergang von einem Laufwerk auf das nächste nicht.

Man merkt also, die Welt der Commodore-Rechner und ihre Umgebung bietet einige Überraschungen, zumal es einigen Eingeweihten eingefallen ist, weitere Chips in Ihren Geräten unterzubringen oder auszutauschen, entweder um sich nervenaufreibende Wartezeiten zu ersparen, oder die Heizleistung Ihrer Laufwerke herabzusetzen, was durch weniger stromfressende CMOS-CHIPs erreicht werden kann, die jedoch keine Exoten-Befehle (Pseudo-OPs) verstehen.

Wir glauben nicht, daß wir alle feinsinnigen Produkt-Varianten bereits aufgespürt haben (verstellte Laufwerke laufen außer Konkurrenz!). Deshalb können Sie uns immer mit Informationen über Neuentdeckungen auf diesem Sektor begeistern. Schließlich wollen wir INPUT 64 für jeden C64 bieten.

Schreiben Sie uns, wenn Ihnen das Abenteuer C64 neue überraschende Erlebnisse beschert hat.

Auf einen Blick: INPUT 64-Betriebssystembefehle

Inhaltsverzeichnis aufrufen	CTRL und i
Hilfsseite aufrufen	CTRL und h
Programm abspeichern	CTRL und s
Bildschirmausdruck	CTRL und b
Titelbild abkürzen	CTRL und q

Laden von Kassette mit LOAD oder SHIFT+RUN/STOP
Laden von Diskette mit LOAD INPUT ,8,1*
Ausführliche Bedienungshinweise finden Sie auf Seite 29.

Leser fragen...

DATEIKASTEN beschleunigt

Warum lassen sich eigentlich DATEIKASTEN-Dateien nicht mit SuperTape 'saven'?
(verschiedene Leser)

Sie lassen sich, allerdings mit zwei Änderungen im BASIC-Listing des Programms DATEIKASTEN (Ausgabe 1/85). Ändern Sie in den Zeilen 4100 und 4600 jeweils die zweite 1 in der Befehlsfolge OPEN1,1,1,DN$ beziehungsweise OPEN1,1,0,DN$ um in eine 7 . Achten Sie aber darauf, daß Sie dabei die Länge des Programm nicht verändern, da sich sonst das Maschinenprogramm im Speicher verschiebt! Also nicht 07 eingeben oder erst die 1 löschen und dann die 7 eingeben, sondern nur die entsprechenden Zeilen 'listen', mit dem Cursor auf die 1 fahren, 7 tippen und RETURN. (d. Red.)

INKEY abgehängt?

Zuerst aufgefallen ist es mir bei dem Programm REISEKOSTENBERECHNUNG (Ausgabe 8/85): Nach dem Laden wird durch LIST nur eine BASIC-Zeile ausgegeben, und erst nach dem Start durch RUN ist das ganze Programm 'listbar'. Was ist da los? (tel. Anfrage)

Dieses 'Erlebnis' können Sie bei vielen INPUT 64-Programmen haben. Grund: die Einbindung der PRINT AT- und INKEY-Routinen aus Ausgabe 6/85. Diese liegen am Anfang des BASIC-Speichers, und das durch den SYS-Aufruf in der ersten Zeile angesprungene Maschinenprogramm setzt den Anfang des BASIC-Bereichs auf das eigentliche Programm und startet dies. Falls Sie also Veränderungen im BASIC-Teil solcher Programme vornehmen, müssen Sie dies kurz starten, mit RUN/STOP anhalten und dann vor dem Abspeichern durch ein POKE 44,8 den BASIC-Anfang wieder heruntersetzen. So werden auch die beiden Routinen mit abgespeichert. (d. Red.)

MANAGER unschlagbar?

Was macht der Manager (im gleichnamigen Programm der Ausgabe 9/85, d. Red.) des 1. FC Köln, wenn er Konzentrationsschwierigkeiten hat? Er schlägt Werder Bremen 11:2, M'Gladbach 13:0, Bayern München 16:0 und verliert anschließend gegen Schalke 14:0. Wie kam es dazu? Ich kaufte mir einen guten Spieler dazu, drückte 6 wie Bilanz, ging wieder ins Hauptmenue. Die Spielstärke meiner Mannschaft steigerte sich immer mehr. Nach meinen Siegen verkaufte ich 13 Einheiten und blätterte zwischen Bilanz und Hauptmenue - bis ich eine Spielstärke von -70 hatte, Schalke deklassierte mich. Folgerung: hier liegt ein Fehler vor! (M. Waßerfuhr, Köln)

Ein Leser soll durch diese Methode sogar ein 20:0 im Spiel Hannover 96 gegen Werder Bremen erzielt haben, und der war begeistert. Im Ernst: es handelt sich nicht um hinterrücks einprogrammierten Lokalpatriotismus, sondern um einen Programmfehler. Laden Sie das Programm (von Ihrem eigenen Datenträger), starten es und drücken die RUN/STOP-Taste. Ein LIST 311 zeigt die fehlerhafte Zeile. Statt 311 ZL=0 muß dort stehen:

Fortsetzung Seite 4

Fortsetzung von Seite 3

311 ZL=0 : JA=0 : JS=0. JA und JS sind die Variablen für die Abwehr- und die Sturmstärke, die vor der Neuberechnung nach dem Kauf oder Verkauf eines Spielers auf Null gesetzt werden müssen. Diese beiden Zuweisungen fielen leider bei der letzten redaktionellen Bearbeitung unter den Tisch.

MOBED gelistet

. . . ist es mir bis jetzt nicht gelungen, die durch den MOBED (Ausgabe 3/85) erzeugten DATA-Zeilen für eine weitere Verwendung sichtbar zu machen. (G. Renauld, Stuttgart)

Wenn Sie die Taste d drücken, erscheint unter dem Editier-Feld der Schriftzug dat , und Sie müssen einen fünfstelligen Namen für Ihr Sprite eingeben. Ein akustisches Signal bestätigt, daß nun die DATA-Zeilen erzeugt sind. Verlassen Sie mit der RUN/STOP-Taste den MOBED und sehen sich per LIST die Zeilen an. Die Sprite-Definition wird immer an die höchsten bereits existierenden Zeilennummern gehängt. Innerhalb des Magazins ist diese Option gesperrt, der MOBED muß also vorher auf Ihren eigenen Datenträger abgespeichert und von dort geladen worden sein.

DISKMONITOR demonstriert

Nach Anwählen dieses Programms erscheint lediglich die Erläuterung der Befehle, die sich ständig wiederholt. Ein Einstieg in den Monitor über SYS 49152 ist nicht möglich, ebensowenig ein Kopieren mit CTRL und s. (N. Haage, Koblenz)

Anscheinend ist dem Autor des Diskmonitors die innerhalb des Magazins laufende Demonstration zu gut gelungen. Bei dem Programm, das Sie beim Anwählen dieses Moduls innerhalb von INPUT 64 starten, handelt es sich lediglich um eine Demonstration der Funktionen des Diskmonitors. Der selbst ist aus technischen Gründen nur außerhalb des Magazins lauffähig, also nach dem Überspielen auf Ihren eigenen Datenträger. Das wiederum muß bei diesem Programm funktionieren, im Hilfsfenster erscheint schließlich die Option Sichern . Manche Versionen der 1541 reagieren allerdings etwas allergisch auf das Zusammenspiel mit dem INPUT;;64-Betriebssystem. Abhilfe schafft immer, die Diskettenstation vor dem Abspeichern einmal kurz aus- und dann wieder einzuschalten.

(d. Red.)

Sternenhimmel im Wohnzimmer:
PLANETARIUM

In den langen und klaren Winternächten lohnt es sich schon einmal, einen Blick auf den nächtlichen Sternenhimmel zu werfen. Der Anblick dieser Milliarden von Sternen fasziniert und verwirrt zugleich. Aber in diesem Gewirr von Sonnen die Übersicht zu behalten, scheint schier unmöglich.

Dieses Programm kann Ihnen nun dabei helfen, Ordnung in dieses vermeintliche Durcheinander zu bringen. Nach Eingabe einiger Werte berechnet es Ihnen den zum angegebenen Zeitpunkt sichtbaren Sternenhimmel und stellt diesen auf dem Bildschirm dar.

Bedienung des Programms

Nach dem Titelbild befinden Sie sich im Hauptmenü. Dort haben Sie nun die folgenden Möglichkeiten:

f1 Erläuterungen

Mit dieser Taste erhalten Sie eine Zusammenfassung dieser Anleitung.

f3 Parametereingabe

Hier haben Sie die Möglichkeit, die Berechnungsgrundlagen für das Programm selbst auszuwählen.

Bei der Eingabe von Monat und Datum achten Sie bitte selbst auf korrekte Werte, und wundern Sie sich bitte nicht, wenn der Computer einen 31. Februar als Eingabe akzeptiert. Wegen der Einführung des gregorianischen Kalenders im Jahr 1583 werden frühere Jahresangaben nicht angenommen. Ansonsten steht Ihnen die Wahl des Datums vollkommen frei!

Zur Eingabe der Uhrzeit gibt es eigentlich nicht viel zu sagen. Hier können Sie jede beliebige Zeit von 00.00 bis 23.59 wählen. Die Zeitangaben beziehen sich auf die mitteleuropäische Zeit (MEZ). Die Sommerzeit ist im Programm nicht berücksichtigt. Für die Umrechnung gilt bekanntlich:

MEZ = MESZ - 1

Als letztes können Sie Ihre Blickrichtung bestimmen. Sie sind dabei nicht auf die vier Himmelsrichtungen angewiesen, sondern können sich auch für Zwischenwerte entscheiden. Dies ist möglich, da die Blickrichtung in Grad eingegeben wird. Jede Eingabe zwischen 000 und 359 Grad wird akzeptiert (360 = 0). Dabei gilt:

000 Grad = Norden
090 Grad = Osten
180 Grad = Süden
270 Grad = Westen

Während der Parametereingabe können Sie die einzelnen Felder direkt mit den Tasten CURSOR UP und CURSOR DOWN anwählen. Wenn Sie die Blickrichtung eingegeben haben, befinden Sie sich wieder im Hauptmenue.

f5 Sternenhimmel

Jetzt wird es interessant: Mit Druck auf diese Taste starten Sie die Berechnungsroutine. Das dauert zwar einige Zeit, aber der Rechner hat nun auch eine ganze Menge zu tun: Er berechnet die Bildschirmkoordinaten für 30 Sternbilder mit insgesamt über 200 Sternen anhand der von Ihnen gewählten Parameter. Nach ungefähr 50 Sekunden ist es dann soweit, und der Sternenhimmel wird zusammen mit Ihren Werten ausgegeben.

Das ist aber noch längst nicht alles. Drücken Sie doch einmal die Leertaste. Wie Sie sehen, zeichnet der Computer nun die zu den Sternen gehörigen Sternbilder ein. Mit nochmaligem Drücken machen Sie diese Darstellung wieder rückgängig. Es gibt im Sternenhimmel-Modus noch weitere Funktionen, von denen Sie eine Übersicht durch Drücken der Taste h bekommen.

Mit den Tasten f3 und f5 können Sie die Farben von Sternen und Himmel Ihrem eigenen Geschmack entsprechend ändern. Mit den Tasten CURSOR DOWN / CURSOR RIGHT ist eine Veränderung der Blickrichtung in Schritten von 5 Grad möglich. Der laufende Wert wird auf dem Bildschirm angezeigt. Mit der Taste f7 verändern Sie den Zoom-Faktor, der Werte zwischen 0 und 9 annehmen kann. (der Normalwert ist 0). Dadurch ist es möglich, Teile des Sternenhimmels genauer zu betrachten.

Haben Sie Ihre Wahl hinsichtlich Blickwinkel und Zoom-Faktor getroffen, drücken Sie die RETURN-TASTE. Die neuen Werte werden damit übernommen, und der neue Sternenhimmel berechnet. Für die Dauer der Berechnung verfärbt sich der Bildschirmrahmen. Mit der

Taste f1 gelangen Sie wieder zurück in das Hauptmenü.

f7 Langzeitsimulation

Genau wie die Sonne gehen auch die Sterne im Osten auf und im Westen unter. Wenn man nun einen Fotoapparat auf Langzeitbelichtung einstellt und mit ihm einen Himmelsausschnitt anpeilt, so erhält man ein Bild, auf dem jeder Stern eine je nach Belichtungsdauer verschieden lange Spur erzeugt. Derartige Fotos kann man mit dieser Funktion simulieren.

Den gewünschten Himmelsausschnitt können Sie wieder mit Hilfe der Parametereingabe festlegen. Daraufhin drücken Sie die Taste f7 und werden nach der gewünschten Simulationsdauer gefragt. Diese ist in Minuten einzugeben. Sie sollten allerdings mit kleinen Werten beginnen, da diese Berechnungen sehr zeitaufwendig sind. Wenn Sie die Simulation abbrechen wollen, genügt es, irgendeine Taste zu drücken. Danach befinden Sie sich wieder im Hauptmenü.

Damit wären dann alle Funktionen des Programmes erklärt. Die Art der Anwendung bleibt Ihnen natürlich selbst überlassen. Sie könnten zum Beispiel:
...den Sternenhimmel am Tag Ihrer Geburt betrachten;...den Winter- und Sommerhimmel miteinander vergleichen;...den Gang der Sterne im Verlauf eines Jahres verfolgen;

Und für all diese Dinge brauchen Sie nie auf gutes Wetter zu warten. Denn Ihr C-64 kennt keine Bewölkung, sondern nur klare Nächte. Klare Tage übrigens auch, denn Sie können sich den Sternenhimmel auch am Tage ansehen. Dieses sollte Sie allerdings nicht dazu bewegen, fortan nur noch vor dem Monitor zu sitzen. Denn einen so richtig schönen Sternenhimmel wird ein Computer wohl nie ersetzen können...

Weißt Du, wieviel Sterne stehen...

Seit der Mensch denken kann, hat er sich immer wieder mit dem Wesen der Sterne beschäftigt. Den Himmel stellte man sich als eine riesige Kuppel vor, die die nach damaliger Vorstellung flache Erde überdacht. Man glaubte, daß die Sterne an dieser Kuppel fixiert seien. Um die Orientierung am Himmel zu erleichtern, wurden die Sternbilder eingeführt, die meist einen mythologischen Bezug besaßen. Sie brauchen gar nicht erst zu versuchen, bei allen Sternzeichen einen Bezug zwischen ihrem Namen und ihrem Aussehen zu finden. Das ist nämlich in den meisten Fällen schlicht und einfach unmöglich. Die Konstruktion der einzelnen Sternbilder ist in keiner Weise festgelegt. So existieren bei den meisten Sternzeichen viele verschiedene Versionen, die einzelnen Sterne mit Linien zu Bildern zu verbinden.

Heutzutage weiß man natürlich besser über die Sterne und ihren Lauf Bescheid. Die Wissenschaft, die sich mit dieser Problematik befaßt, heißt Astronomie (nicht Astrologie - der Unterschied ist gewaltig!). Heute weiß man, daß die Erde keine Scheibe ist, und daß die Sterne nichts weiter sind, als sonnenähnliche Himmelskörper, die sich irgendwo im unendlichen Weltraum befinden. Was geblieben ist, sind die Sternbilder, die uns auch heute noch die Orientierung am Himmel erleichtern. Auch eine andere Vorstellung wurde teilweise Übernommen: Nimmt man an, daß die Sterne sich nicht wie in Wirklichkeit im unendlichen Raum befinden, sondern auf einer gedachten kugelförmigen Kuppel, die die Erde umgibt, lassen sie sich mit denselben Mitteln kartographisch erfassen wie Gebiete auf der Erdoberfläche.

Genau das wird bei der Konstruktion von Sternkarten vorausgesetzt. Diese gedachte Himmelskugel besitzt einen Nordpol, dessen Lage beinahe identisch ist mit der des Polarsterns. Analog dazu existiert natürlich auch ein Himmelssüdpol, der sich nahe beim Sternbild Kreuz des Südens befindet. Der in den Weltraum projizierte Erdäquator schließlich wird zum Himmelsäquator. Um die Position eines Sternes eindeutig bestimmen zu können, bedarf es nun eines Systems ähnlich dem der Längen- und Breitengrade auf der Erdoberfläche. Deklination und Rektaszension heißen die entsprechenden Begriffe, mit denen man jedem Punkt am Himmel einen Positionswert zuordnen kann. Die Deklination entspricht der geografischen Breite. Sie kann Werte von +90 bis -90 annehmen. Die Rektaszension entspricht der geografischen Länge. Sie kann Werte von 0 bis 360 Grad annehmen.

Wie man Sternpositionen bezeichnet, wissen wir jetzt. Bei der Darstellung der Himmelsobjekte auf Sternkarten gibt es allerdings die gleichen Probleme wie bei Landkarten der Erdoberfläche, wenn man versucht, eine gekrümmte Fläche auf eine ebene Fläche (Landkarte) zu projizieren. Verzerrungen lassen sich dabei nicht vermeiden.

Von allen Sternkarten, die es gibt, ist die dreh-

bare sicher die interessanteste. Viele werden sie schon einmal in der Hand gehabt haben, ohne allerdings zu wissen, wie man mit ihr umgeht. Für diejenigen, die eine drehbare Sternkarte oder deren Handhabung nicht kennen, wollen wir diese kurz beschreiben.

Sie besteht aus einer runden Sternkarte, auf der ein bewegliches Sichtfenster drehbar angebracht ist. Dieses Sichtfenster muß man Datum und Uhrzeit entsprechend anhand einer angebrachten Skala einstellen, so daß im Sichtfenster der momentan sichtbaren Sternenhimmel beobachtet werden kann. Meist gibt es noch einige Zusatzfunktionen, die hier aber nicht weiter erläutert werden sollen.

Die Funktion einer solchen Sternkarte wird vom Programm PLANETARIUM simuliert. Dazu sind die Koordinaten aller dargestellten Sterne im Format Rektaszension / Deklination im Programm abgespeichert. Die durch Datum und Uhrzeit bedingte, je nach Entfernung vom Himmelspol verschieden große Abweichung läßt sich nun mathematisch erfassen. Nach der entsprechenden Berechnung erhält man dann die Koordinaten für den jeweils aktuellen Sternenhimmel. Als nächstes berechnet das Programmden in der gewählten Blickrichtung tatsächlich sichtbaren Himmelsausschnitt. Zum Schluß werden dann die erhaltenen Werte noch in Bildschirmkoordinaten umgerechnet und auf Ihrem Monitor dargestellt.

Bei der Langzeitsimulation wird der aktuelle Sternenhimmel der gewählten Simulationsdauer entsprechend in Zeitabschnitten von drei Minuten immer wieder neu berechnet und auf dem Bildschirm ausgegeben. So entstehen bei längeren Simulationszeiten die Bahnspuren, wie man sie auch bei Langzeitfotografien beobachten kann.

Hinweise zum Programm:

Die Berechnung beschränkt sich auf den nördlichen Sternenhimmel. Auf eine wählbare Eingabe des Beobachtungsortes wurde ebenfalls verzichtet. Das Programm geht davon aus, daß Sie sich im deutschsprachigen Raum befinden. Innerhalb dieses Gebietes hält sich die Verschiebung der Koordinaten durch einen anderen Standort in vertretbaren Grenzen und fällt kaum ins Gewicht. Es kann aber zu kleinen Ungenauigkeiten kommen, weil zum Beispiel nur ein Teil aller wirklich sichtbaren Sterne berücksichtigt werden konnte und weil die Umrechnung der Sternkoordinaten in Bildschirmkoordinaten Verzerrungen mit sich bringen kann.

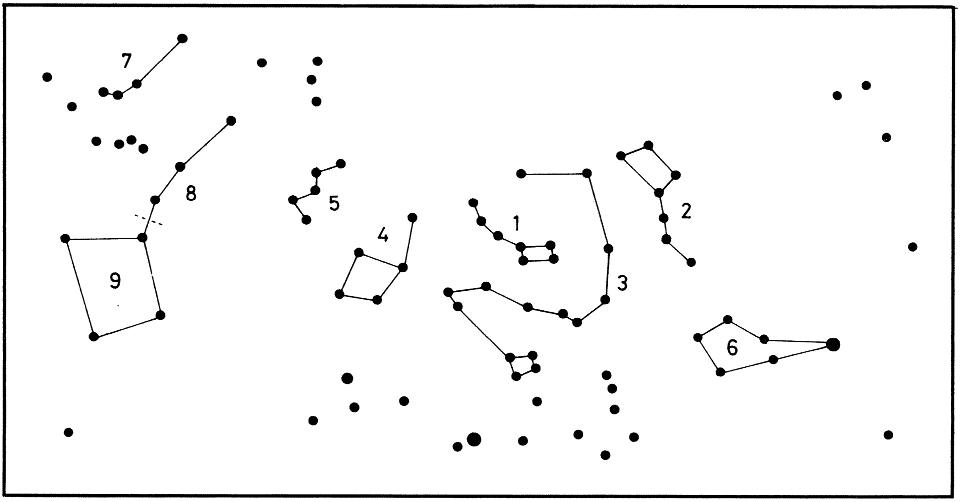
Sie sollten beachten, daß Sie das Programm PLANETARIUM nur direkt nach dem Laden mit der üblichen Tastenkombination CTRL und s aus INPUT 64 heraus abspeichern können. Zu einem späteren Zeitpunkt ist dies nicht mehr möglich, da sich das Programm selbst verschiebt.

Orientierungshilfen

Wir haben im folgenden einige Bildschirm-Hardcoppys aufgeführt. Die Parameter wurden so gewählt, daß jedes Sternbild einmal vollständig zu erkennen ist.

alphabetische Aufstellung der Sternbilder

Sternbilder	Beispiel	Nummer
Adler	5	25
Andromeda	1	8
Bootes	1	6
Cepheus	1	4
Drache	1	3
Fische	3	19
Fuhrmann	3	16
großer Bär	1	2
großer Hund	2	14
großer Wagen	1	2
Herkules	7	30
Jungfrau	4	21
Kassiopeia	1	5
kleiner Bär	1	1
kleiner Hund	2	13
kleiner Wagen	1	1
Krebs	2	12
Leier	7	29
Löwe	4	22
Orion	2	15
Pegasus	1	9
Plejaden	2	10
Schütze	6	26
Schwan	7	28
Siebengestirn	2	10
Skorpion	6	27
Steinbock	5	24
Stier	3	17
Waage	4	20
Wassermann	5	23
Widder	1	7
Zwillinge	2	11

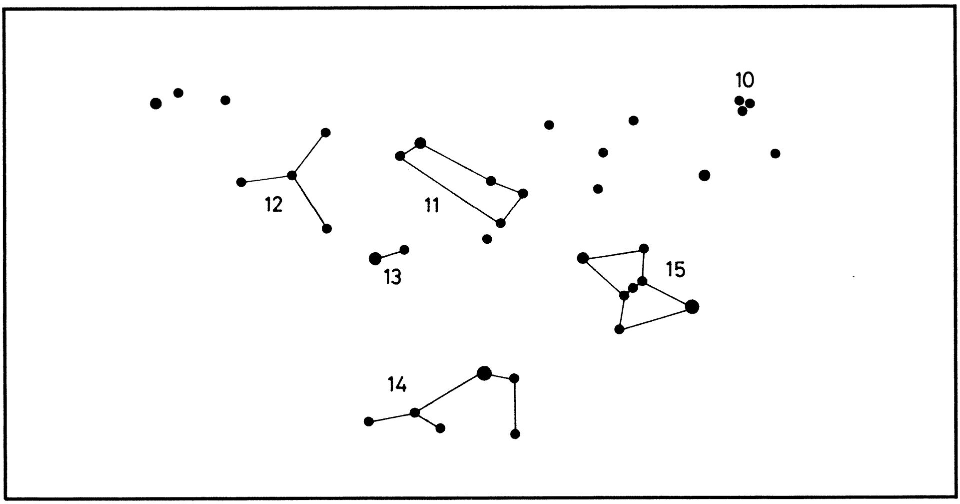


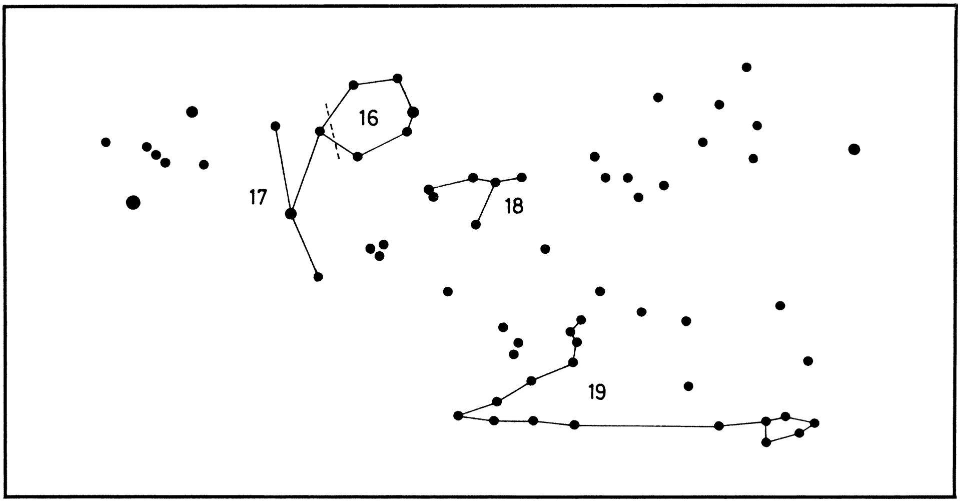
Beispiel 1:

Kleiner Bär oder kleiner Wagen (1), großer Bär oder großer Wagen (2), Drache (3), Cepheus (4), Kassiopeia (5), Bootes (6), Widder (7), Andromeda (8), Pegasus (9)

Beispiel 2:

Plejaden oder Siebengestirn (10), Zwillinge (11), Krebs (12), kleiner Hund (13), großer Hund (14), Orion (15)



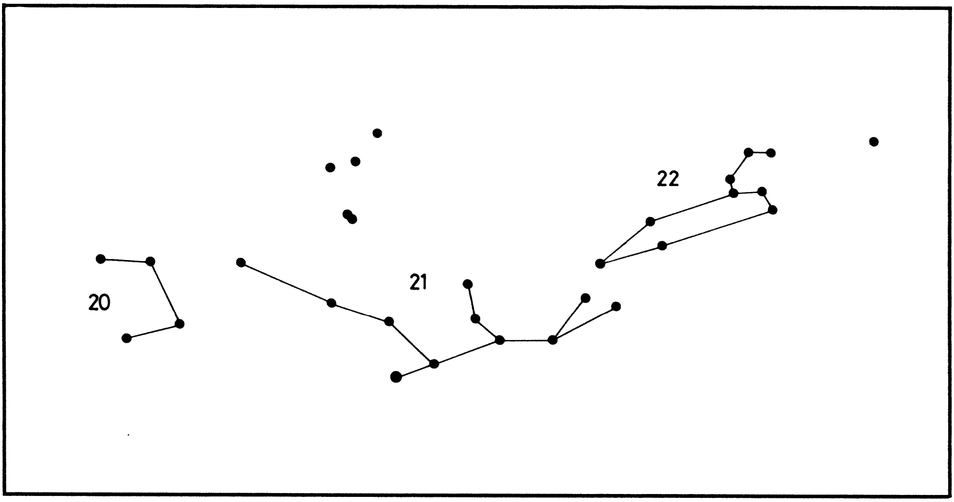


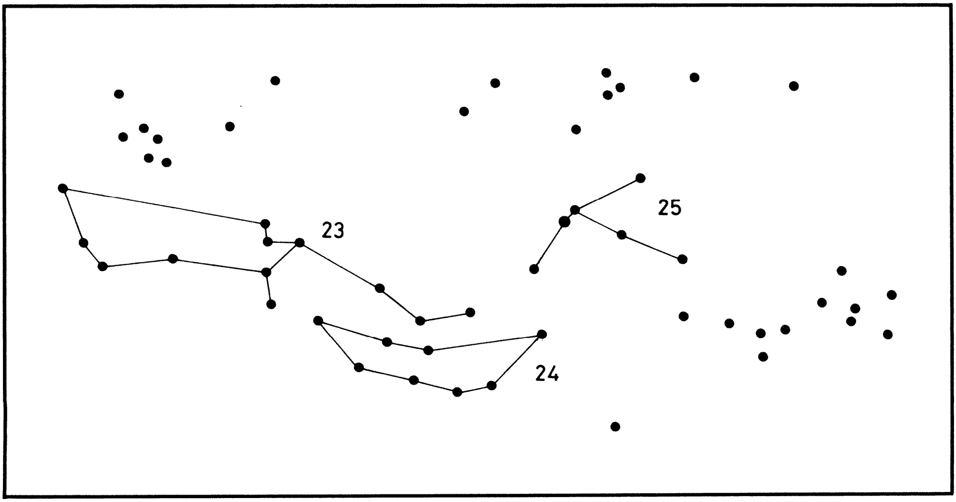
Beispiel 3:

Fuhrmann (16), Stier (17), Perseus (18), Fische (19)

Beispiel 4:

Waage (20), Jungfrau (21), Löwe (22)



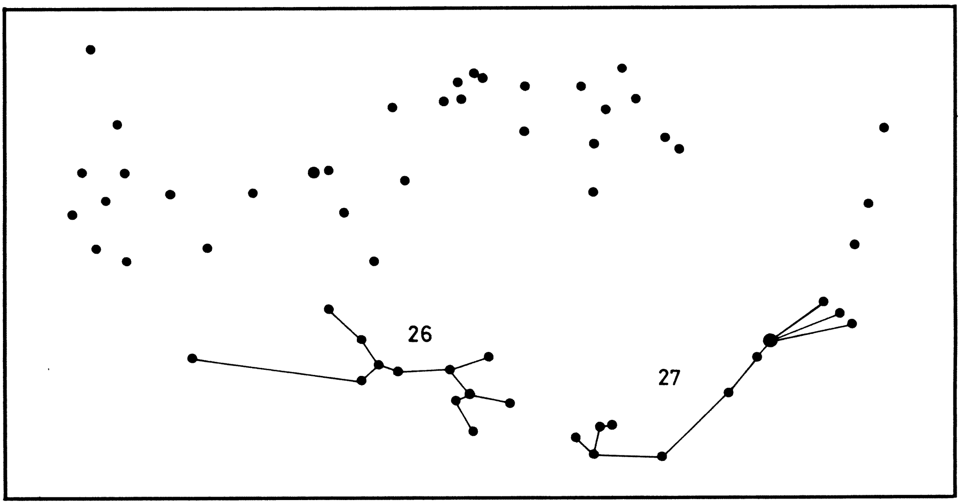


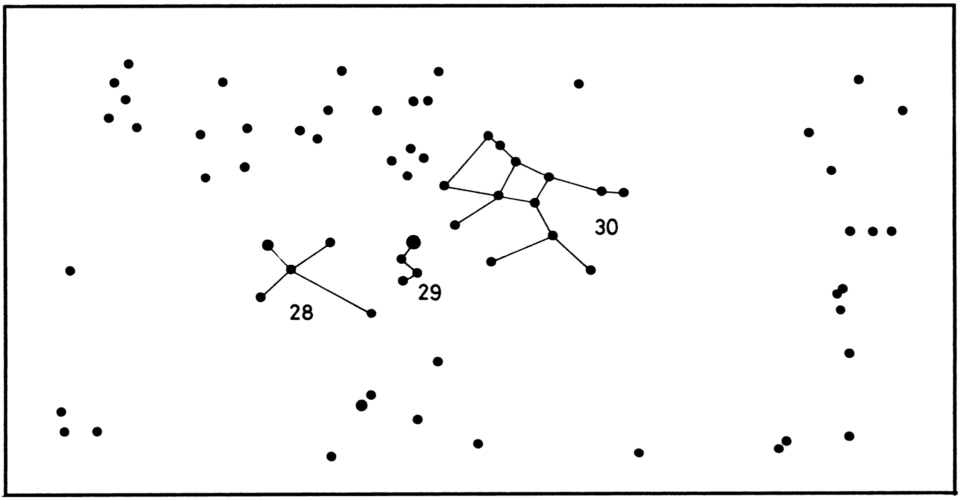
Beispiel 5:

Wassermann (23), Steinbock (24), Adler (25)

Beispiel 6:

Schütze (26), Skorpion (27)





Beispiel 7:

Schwan (28), Leier (29), Herkules (30)

Literaturhinweise:

dtv-Atlas zur Astronomie Deutscher Taschenbuch Verlag ISBN 3-423-03006-2

Das Himmelsjahr - Sonne, Mond und Sterne im Jahreslauf Franckh'sche Verlagshandlung Stuttgart ISBN 3-440-05348-2 (Erscheint für jedes Jahr mit aktuellen Daten neu.)

Drehbare Kosmos-Sternkarte Franckh'sche Verlagshandlung Stuttgart ISBN 3-440-04916-7

Hilfsprogramm
Basic-Zeilen Tool

Diesmal bieten wir Ihnen ein BASIC-Zeilen-Tool. Mit einem einfachen SYS-Aufruf können Sie innerhalb eines laufenden BASIC-Programms BASIC-Zeilen erzeugen. Wie üblich bieten wir Ihnen zwei Versionen, eine liegt im freien RAM-Bereich hinter dem Bereich, den SuperTape beansprucht, die andere liegt am BASIC-Anfang und setzt den BASIC-Start automatisch hoch. Haben Sie anschließend ein BASIC-Programm eingegeben, müssen Sie vor dem abspeichern mit POKE 43,0 und POKE 44,8 den BASIC-Start zurücksetzen.

Syntax. . .

SYS Adresse,Flag,String

Adresse: je nach Version Startadresse des Tools, also 51200 ($C800) beziehungsweise 2816 ($0B00).

Flag: '0' = neue Zeile eingefügt
'1' = Zeile überschrieben
Flag muß eine BASIC-Variable sein!

String: enthält den Text der BASIC-Zeile mit Zeilennummer.

. . . und Anwendung

Diese Möglichkeit ist zum Beispiel für Programme interessant, mit denen verschiedene mathematische Funktionen berechnet werden sollen. Sie können das BASIC-Zeilen-Tool in Verbindung mit dem INPUT-Befehl verwenden, indem Sie die gewünschte Formel über INPUT eingeben und den String anschließend an den SYS-Aufruf übergeben.

Zum Beispiel:

```
100 INPUT BI$
110 BZ$= "210 : Y= "+BI$
120 SYS 52000,F,BZ$
130 PRINT CHR$(147); "FUER Y= ";BI$
140 PRINT "X ", "Y "
150 PRINT
200 FOR X=1TO10
220 : PRINT X,Y
230 NEXT X
240 END
```

Dieses kleine Programm würde zu einer eingegebenen Formel jeweils das Ergebnis für X-Werte von 1 bis 10 ausgeben. Nach dem ersten Aufruf des Programms enthält das Programm zusätzlich die Zeile 210, beispielsweise:

```
210 : Y=SIN(X)*X ↑2+.45*X
```

Die Variable F enthält den Wert '0', da eine neue Zeile eingefügt wurde. Wird das Programm erneut mit RUN gestartet, bekommt F den Wert 1 zugewiesen, da die Zeile 210 überschrieben wurde.

Denken Sie bitte daran, daß der BASIC-Interpreter die Zeile 210 ganz normal abarbeitet und dadurch auch die üblichen Fehlermeldungen auftreten können, die zum Programm-Abbruch führen. Beispiele sind: DIVISION - BY ZERO ERROR oder SYNTAX ERROR

Selbstverständlich können Sie die BASIC-Zeile auch direkt als String mit Hochkommas eingeben:

```
SYS 51200,F, "210 Y=2*X/COS(X) "
```

Sogar PRINT-Befehle mit Strings sind möglich:

```
SYS 51200,F, "400 ? "TEST " "
```

wobei Sie auch die üblichen Abkürzungen für BASIC-Befehle verwenden können.

Bei welcher Programmstruktur der Einsatz solcher Befehle sinnvoll ist, überlassen wir Ihrer Phantasie. Besonders interessante und neuartige Ideen können Sie uns gerne zusenden. Vielleicht kann das eine oder andere Programm in unserer ID-Werkstatt veröffentlicht werden.

Physik mit Nico

Die angekündigten physikalischen Fragestellungen waren für den kleinen Nico wohl doch zu schwer; er fällt nämlich immer auf die Nase. Damit Ihnen dieses nicht auch passiert, werden die Lösungswege wieder ausführlich erklärt. Ihnen steht selbstverständlich auch diesmal wieder eine Rechenseite zur Verfügung, auf der Sie alle Rechenoperationen ausführen können, die Ihnen vom Direkt-Modus her bekannt sind. Na dann beschleunigen Sie mal schön. . . .

SUPER MEMORY

Super Memory ist ein Denkspiel für eine Person. In einem vier × vier Kästchen großen Feld, werden 16 Ziffern dargestellt. Ihre Aufgabe ist es, die Ziffern systematisch so zu ändern, daß zum Schluß in jedem Kästchen eine Neun steht.

Das Ändern der Ziffern übernimmt der Rechner. Sie brauchen nur auf die jeweils zugeordnete Taste zu drücken. Die Tastatur-Belegung zeigt Ihnen der Rechner auf dem Bildschirm an (Bild 1). Allerdings wird nicht nur der Wert der angewählten Ziffer um eins erhöht, sondern der Wert aller Ziffern, die in der gleichen Spalte und in der gleichen Zeile liegen. Eine Neun wird durch das Hochzählen zu einer Null.

So funktioniert's

Nach dem Start des Spiels und einigen Hinweisen erscheint ein kleines Menü. Hier können Sie wählen, ob Sie mit oder ohne Zeit-Limit spielen möchten. Zu Anfang ist es schon angenehmer, wenn man nicht unter Zeitdruck spielen muß.

Haben Sie gewählt, sehen Sie im linken Drittel des Bildschirmes das Tastaturbelegungs-Feld und im mittleren Drittel das Zahlenfeld. In der unteren Bildschirmhälfte werden die Zeit, die Punkte und die Anzahl der Züge angezeigt. Geben Sie jetzt bitte ein, mit wieviel Schritten Sie das Endergebnis erreichen wollen. Die Schrittzahl darf nicht kleiner als zwei und nicht größer als neunundneunzig sein.

Ein Beispiel

Hat der Rechner Ihnen ein Zahlenfeld vorgegeben (Bild 2), drücken Sie in unserem Beispiel die Taste 'H'. Nicht nur die korrespondierende 8, sondern auch alle Ziffern der gleichen Spalte und der gleichen Zeile werden nun um eins erhöht. Das Ergebnis zeigt Bild 3.

Ein Druck auf die Taste 'I' (Bild 3) inkrementiert alle Ziffern der Zeile zwei und der Spalte vier. Es erscheint in jedem Kästchen eine Neun. Das Spielziel ist erreicht.

Übrigens...

...mit der Funktionstaste F7 können Sie, falls Sie sich einmal verrannt haben, das Zahlenfeld wieder in seinen Ausgangszustand zurückversetzen.

Mit der Pfeil-links-Taste (oben links auf der Tastatur), können Sie Ihren letzten Zug wieder zurücknehmen.

Möchten Sie eine Pause machen, drücken Sie einfach auf die Leertaste.

Wenn Sie das Handtuch werfen, also aufgeben wollen, signalisieren Sie dem C64 dieses mit der Funktionstaste F1. Aber das wird Ihnen doch nicht passieren – oder...?

5	6	7	8
T	Y	U	I
G	H	J	K
V	B	N	M

Tastaturbelegungs-Feld (Bild 1)

9	8	9	8
8	7	8	8
8	8	8	7
9	8	9	8

Nach Drücken der Taste H... (Bild 2)

9	9	9	8
8	8	8	8
9	9	9	8
9	9	9	8

...ist das Spielziel fast erreicht. (Bild 3)

SID-Kurs Teil 6

Anschließend an Teil 5 geht es diesmal um den letzten Schliff bei der Programmierung von Rhythmuseffekten und Bassläufen. Zwei Beispielprogramme können Sie durch CTRL und s auf Ihren eigenen Datenträger überspielen. Denken Sie bei der Eingabe eigener Noten daran, daß die letzte Zeile der DATA-Statements das Endekennzeichen enthalten muß!

Listing Beispielprogramm No. 1

```
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+7
20 poke54277+7,1*16+4:poke54278+7,6*16+7:poke54296,15:poke54272+17,8
30 poke54291,2*16+8:poke54292,10*16+10
100 readl1,h1:poke54276,32 :poke54276+14,64
105 ifl1=-2thenpoke54286,h1:readb,l1,h1:poke54287,b:poke54290,65
110 ifh1<>0thenpoke54276,33:poke54272,l1:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then140
130 poke54276+7,33:poke54272+7,l2:poke54273+7,h2
140 forq=1to200:nextq:ifl1<>-1then100
500 rem datas
510 data 180,8, 180,8, -1,0, 0,0
```

Listing Beispielprogramm No. 2

```
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+7
20 poke54277+7,1*16+4:poke54278+7,6*16+7:poke54296,15:poke54272+17,8
30 w=33
100 readl1,h1:poke54276,w-1:poke54276+14,64
105 ifl1=-3thenw=17:goto100
110 ifh1<>0thenpoke54276,w :poke54272,l1:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then140
130 poke54276+7,33:poke54272+7,l2:poke54273+7,h2
140 forq=1to200:nextq:ifl1<>-1then100
500 rem datas
510 data 180,8, 180,8, -1,0, 0,0
```

DatCopy

Die Idee zu diesem Programm entstand aufgrund mehrerer Leseranfragen: Wie können DATEIKASTEN-Dateien von Kassette auf Diskette kopiert werden und umgekehrt? (DATEIKASTEN ist ein Dateiverwaltungsprogramm, das in Ausgabe 1/85 veröffentlicht wurde.) Natürlich kopiert DatCopy nicht nur Dateikasten-Dateien, sondern jede sequentielle Datei, und zwar von den Aufzeichnungsverfahren 1541-Diskette, Commodore-Kassette und SuperTape DII-Kassette[1] auf jedes dieser Formate.

DatCopy ist überwiegend in BASIC geschrieben, für zeitkritische Operationen werden Maschinensprache-Routinen benutzt; zum Beispiel die für das Lesen der Daten von Band oder Diskette und die Zwischenspeicherung in einer Tabelle im Rechner. Diese Routinen belegen den von BASIC nicht benutzten Bereich von Adresse 52000 ($CB20) bis 53247 ($CFFF). Für die Zwischenlagerung der gelesenen Daten dienen die Adressen 8000 ($1F40) bis 40000 ($9C40). Die Maximallänge einer Datei beträgt so exakt 31999 Bytes – das dürfte wohl immer ausreichen. Um Kollisionen zwischen Programm-Variablen und Daten zu vermeiden, werden zu Anfang des Programms die Zero-Page-Adressen 45/46 ($2D/$2E), die den Start der Variablen festlegen, auf das Ende der Datentabelle 'gepoked'. Es ist deswegen nicht sinnvoll, DatCopy nach dem Start noch einmal abzuspeichern – es sei denn, DatCopy ist Ihnen 38 KByte auf Ihrem Band wert (entspricht 152 Blöcken auf der Diskette).

Als erste Eingabe wird die Entscheidung zwischen Dateikasten-Dateien und sonstigen Dateien angefordert. Diese Unterscheidung ist notwendig, da Dateikasten-Dateien ein anderes Lese- und Schreib-Verfahren als sonstige sequentielle Dateien benötigen. (Für Detail-Interessierte: Normalerweise können keine Null-Bytes aufs Band geschrieben werden, da diese beim Lesen als Ende-Kennzeichen interpretiert werden. Die Dateikasten-Routinen werten diese Null-Bytes als gültige Daten.) Die Abfrage nach Quelle und Ziel können Sie mit s (SuperTape DII), c (Commodore) oder d (Diskette) beantworten. Für die Option SuperTape - DII muß dieses schon vor dem Start von DatCopy geladen und initialisiert sein. Ansonsten erscheint in der weiß geschriebenen Statuszeile die Meldung SuperTape DII fehlt! . Apropos Fehlermeldungen: Diese werden immer in der Statuszeile ausgegeben, ebenso die notwendigen Hinweise zur Bedienung, beispielsweise Diskette einlegen , Press play on tape und so weiter.

Es gibt eine todsichere Methode, das Programm zum Absturz zu bringen: Wenn nämlich bei der Suche nach Commodore-Dateien eine SuperTape DII-Datei gefunden wird. Aber Mischkassetten kommen ja bei ordentlichen Menschen ohnehin nicht vor. Falls Sie als Ziel der Konvertierung Commodore-Kassette gewählt haben, bedenken Sie, daß Dateien in diesem Format etwa den zwölffachen Platzbedarf auf dem Band haben wie SuperTape DII-Dateien. Also genügend langes Band wählen, denn hochgesprungene Recorder-Tasten werden vom Programm nicht erkannt.

zu (1): SuperTape ist ein schnelles Kassetten-Aufzeichungsverfahren, das erstmals in der Ausgabe 3/84 unserer Schwesterzeitschrift c't vorgestellt wurde. SuperTape ist für die gängigsten Home-Computer implementiert, so daß eine Datenübertragung zwischen verschiedenen Rechner-Typen möglich ist. SuperTape DII für den C 64 lädt und speichert sowohl Programm- als auch sequentielle Dateien. Mit einer Geschwindigkeit von 3600 Baud ist es zwölfmal schneller als das Commodore-Verfahren. SuperTape DII wurde in INPUT 64 Ausgabe 4/85 veröffentlicht (kann nachbestellt werden) und ist außerdem über den Heise-Software-Service erhältlich.

ROBBY

Sie müssen dem Roboter den Weg bahnen. Auf der Suche nach Schrauben, Ölkannen, Zahnrädern und anderen Teilen führt der Weg durchs Labyrinth, das Sie so verschieben müssen, daß sich ein durchgehender Weg ergibt.

Nach dem Titelbild erscheint das Labyrinth auf dem Bildschirm. Oben links am Eingang steht der Roboter und wartet auf einen Tastendruck von Ihnen.

Fortsetzung Seite 16

Fortsetzung von Seite 15

Das Labyrinth besteht aus einem unbeweglichen Teil - dem äußeren Rahmen - und aus einem beweglichen Teil. Im unbeweglichen Teil liegen die Gegenstände, die der Roboter auf seiner Reise aufsammeln soll. Der bewegliche Teil besteht aus mehreren gleichgroßen Segmenten, die so verschoben werden müssen, daß der Roboter einen Durchgang findet. Bei Kreuzungen läuft der Roboter immer geradeaus.

INPUT 64 Leser kennen die Steuerung schon aus Paulchens Schiebung in unserer Ausgabe 8/85. Sie können die einzelnen Labyrinth-Teile entweder mit dem Joystick in Port 2 oder oder über die Tasten 'A', 'Z', ',', '.' steuern, wobei gilt:

Taste:	schiebt Teil nach:
A	oben
Z	unten
,	links
.	rechts

Leertaste oder Feuerknopf startet den Roboter.

Für jeden erreichten Gegenstand gibt es 10 Punkte. Sammelt der Roboter mit Ihrer Hilfe die Gegenstände in weniger als 3 Minuten, gibt es Zusatzpunkte.

Haben Sie mehr als 499 Punkte erreicht, können Sie in der nächsten Spielrunde mit einem zusätzlichen Roboter weiterspielen.

Und nun viel Spaß beim Umgang mit Robotern.

64er Tips

Kennen Sie die Routinen des Betriebssystems? Wußten Sie schon, daß es eine Reihe nützlicher ROM-Routinen gibt, die von BASIC aus mit einem SYS-Befehl direkt aufgerufen werden können?

Wir stellen Ihnen in den 64er Tips einige solcher Routinen vor. Dies sind Maschinen-Programme, die das Betriebssystem benutzt, wenn Sie mit Ihrem Rechner arbeiten. Da das Betriebssytem und der BASIC-Interpreter im ROM Ihres Rechners fest installiert sind, stehen diese Programme immer zur Verfügung, es sei denn, Sie haben den RAM-Bereich eingeschaltet und damit die Benutzung der ROMS unterbunden. Einige der Routinen ersparen einem die umständliche Programmierung von BASIC-Lösungen. Warum also erst Programmteile schreiben, die sowieso schon im Rechner eingebaut sind.

Wir denken, daß wir Ihnen in einer der nächsten Folgen weitere Routinen vorstellen können. Bei der Diskussion über dieses Thema entstand in der Redaktion die Überlegung, von Zeit zu Zeit Profi-Tips für Experten anzubieten. Wir wüßten gerne, ob Sie als Leser Interesse an einer solchen Rubrik hätten. Sie können uns dazu schreiben, falls Sie Zeit und eine Briefmarke übrig haben.

DiscDoctor

Nichts ärgerlicher, als ein wichtiges Programm versehentlich gelöscht zu haben. Oder: beim Abspeichern auf Diskette mit dem 'Klammeraffen' (@) wurden die falschen Files überschrieben - ein mittlerweile bekannter Fehler der 1541-Firmware. Mit genügenden Kenntnissen des DOS (Disc-Operating-System) und der Daten-Organisation auf der Diskette lassen sich natürlich viele Fehler wieder ausbügeln. Das Programm DiscDoctor erledigt die meisten dieser Reparaturen fast automatisch - ohne vom Anwender zu verlangen, sich erst einmal in die Theorie von DOS, BAM, Tracks und Sektoren einzuarbeiten.

Das Anfangs-Menue zeigt die neun zur Verfügung stehenden Optionen des DiscDoctors:

- Einlesen einer neuen Directory
- Suchen nach Files auf der Floppy
- Verfolgen eines Blocks mit BAM-Korrektur
- Suchen nach Zeichenfolge auf Disk
- Zurückverfolgen eines Blockes
- Generieren einer neuen Directory
- Supersuch Option
- Filenamen ändern
- Hexdump eines Blocks

Bedienung . . .

Die Auswahl erfolgt über die Cursor-Tasten und Bestätigung durch RETURN. Wichtig ist, nach dem Einlegen der zu bearbeitenden Diskette immer zuerst die neue Directory einlesen zu lassen, damit das Programm mit diesen Parametern weiterarbeiten kann.

Der weitere Verlauf der Sitzung hängt von der 'Krankheit' der Diskette ab. Angenommen, Sie haben nur ein File versehentlich durch den SCRATCH-Befehl gelöscht. Dann ist Ihnen bestimmt schon beim Einlesen der Directory aufgefallen, daß auch das gelöschte File angezeigt wurde. Wählen Sie jetzt 'Verfolgen eines Blocks mit BAM-Korrektur' und geben Track und Sektor des ersten Blocks des gelöschten Files ein (wird beim Einlesen der Directory neben dem Programmnamen angezeigt). Wenn Sie jetzt das Unterprogramm 'Generieren einer neuen Directory' ausführen lassen, ist Ihr Programm schon gerettet. Der Programmteil 'Directory generieren' sollte immer zum Abschluß aufgerufen werden, um die wiederentdeckten Daten auch in das Inhaltsverzeichnis der Diskette aufzunehmen.

Nehmen wir einen schlimmeren Fall an: der Eintrag des 'gescratchten' Files in der Directory wurde bereits durch einen neuen Eintrag überschrieben. Versuchen Sie es dann mit dem Punkt 'Suche nach Files auf der Floppy'. Unterschieden wird zwischen der Suche nach BASIC-Programmen und der nach Maschinensprache-Files. Bei letzteren müssen Sie eine Anfangsadresse (dezimal oder hexadezimal) eingeben; die Startadresse von BASIC-Programmen wird mit 2049 ($0801) angenommen. Dies trifft übrigens bei Programmen, die Sie aus unserem Magazin heraus abgespeichert haben, in der Regel nicht zu! Zur Suche nach solchen Programmen weiter unten. Wurden die gesuchten Programme gefunden, wird dies durch die Angabe von Track und Sektor der Programmanfänge angezeigt. Schreiben Sie sich diese Werte auf. Über den Aufruf von 'Hexdump' können Sie einen Blick auf diese Blöcke werfen (Bildschirmausgabe stoppen: Leertaste) und entscheiden, ob diese Programme in die Directory aufgenommen werden sollen. In die BAM (Block Available Map = Tabelle der verfügbaren Blöcke) werden die gefundenen Files automatisch eingetragen. In der neu erzeugten Direktory werden zunächst nur die Namen 'PGM Nr.' vergeben, eine Namensänderung ist leicht möglich durch den Aufruf des Menue-Punktes 'File-Namen ändern'. Die bestehenden Namen werden der Reihe nach angezeigt. Geben Sie entweder einen anderen Namen ein oder blättern mit RETURN weiter zum nächsten Namen. Auch diese veränderten Namen müssen durch 'Neue Directory generieren' auf die Diskette zurückgeschrieben werden. Achten Sie darauf, keine Namen doppelt zu vergeben, das Programm kontrolliert dies nicht. Sie können natürlich problemlos einen falschen Namen erneut ändern.

Nehmen wir einen noch schlimmeren Fall an: die Anfangsadresse des verlorenen Programms ist unbekannt. Lassen Sie sich dann etwas Zeit für die 'Supersuch-Option'. Diese Routine liefert die Track- und Sektor-Nummern aller möglichen Programmanfänge. Das heißt, alle Blöcke werden daraufhin untersucht, ob sie Nachfolger eines anderen Blocks sind, ob sie überhaupt Daten enthalten und ob sie bereits in der BAM markiert sind. Stellt das Programm fest, daß ein Block kein Leerblock, nicht Nachfolger eines anderen Blocks und noch nicht in der BAM als belegt gekennzeichnet ist, wird dieser Block als möglicher Programmanfang angenommen. Sind alle Blöcke daraufhin untersucht, können Sie entscheiden, ob diese in die BAM und die Directory aufgenommen werden sollen.

Eine andere Möglichkeit: der Anfang eines wichtigen Programmes ist überschrieben, Sie wollen aber trotzdem eventuell übriggebliebene Daten retten. Suchen Sie dazu zunächst irgend einen Teil Ihrer Daten, und zwar über die Funktion 'Suche nach Zeichenfolgen auf Disk'. Gesucht werden kann nach Textausdrücken, zum Beispiel aus PRINT-Zeilen, String-Zuweisungen oder Kommentaren, nicht nach BASIC-Befehlen, weil diese als TOKEN auf der Diskette abgelegt sind. (Zum Begriff TOKEN siehe die Erläuterungen zur Rätselauflösung in diesem Heft.) Wurde der Ausdruck gefunden, werden wieder Track und Sektor des betreffenden Blocks angezeigt. Kontrollieren Sie über den Aufruf von 'Hexdump', ob es sich wirklich um die gesuchten Daten handeln könnte. Wenn ja, geht es weiter mit 'Zurückverfolgen eines Blocks'. Dieser Programmteil sucht in den benachbarten Blöcken nach Zeigern auf den eingegebenen Block, fragt also, ob diese Vorgänger des jeweiligen Blocks sind. Gesucht wird zunächst im Umkreis von zwei Spuren, da weiter auseinanderliegende verbundene Blöcke selten vorkommen. Sie können aber auch die ganze Diskette durchsuchen lassen. Die durch die Vorgängersuche gefundenen Blocks werden markiert und durch den Aufruf von 'Neue Directory generieren' als neues Programm im Directory verzeichnet. Dadurch werden dann auch alle zum Programm gehörigen Blöcke (also alle Nachfolger) endgültig in der BAM markiert.

. . . und Programmatisches

Durch geeignete Kombination der Möglichkeiten dürften Sie die meisten Disketten-Fehler künftig im Griff haben. Der DiscDoctor ist überwiegend in BASIC geschrieben und wurde durch einige kleine Maschinensprache-Routinen ergänzt. Da diese Routinen wegen ihrer allgemeinen Verwendbarkeit für Assembler-Einsteiger nicht uninteressant sein dürften, noch einige Bemerkungen zu den abgedruckten Listings dieser Routinen.

GET liest einen Block (255 Zeichen) von Diskette in eine Tabelle, die bei 'Anfang' beginnt. Das Y-Register dient als Zeiger auf die aktuelle Tabellenspitze. Die Routine setzt voraus, daß ein Kanal mit der Kanal-Nummer 2 vorher geöffnet wurde. CHKIN legt die Eingabe (BASIN) auf den im X-Register übergebenen Kanal um, CLRCH setzt Ein- und Ausgabe zurück auf Tastatur bzw. Bildschirm.

FIND durchsucht den Speicher nach einer maximal 255 Byte langen Zeichenfolge. In den Adressen 828/829 ($033C/$033D) wird die Anfangsadresse des zu durchsuchenden Bereichs im Low-High-Byte-Format erwartet, in den Adressen 833/834 ($0341/$0342) die Endadresse. Adresse 831 ($033F) enthält die Länge des Suchworts, das ab Adresse 835 ($0343) abgelegt ist. Adresse 830 ($033e) liefert den Wert Null zurück für 'nicht gefunden' oder den Wert 255 für 'gefunden', die Adressen 828/829 enthalten dann die um eins erhöhte Adresse des in der Tabelle vorhandenen Worts.

HEX gibt den Inhalt eines Blocks, dessen Anfangsadresse in den Speicherstellen 900/901 ($0383/$0385) übergeben werden muß, hexadezimal und, falls darstellbar, als ASCII-Zeichen auf dem Bildschirm aus. In den Adressen 902/903 erwartet die Routine HEX die Endadresse +1 des auszugebenden Bereichs.

Assembler-Listing FIND

```
profi-ass 64 v2.0    seite 1

20:   3e00                      .opt p,oo
30:   3e00                      *=   $3e00
40:   3e00 ad 3c 03 find        lda  828    ;anfangsadresse einlesen
40:   3e03 85 03                sta  3
40:   3e05 ad 3d 03             lda  829
40:   3e08 85 04                sta  4
50:   3e0a a9 00                lda  #0
50:   3e0c 8d 3e 03             sta  830    ;zeiger null setzen
60:   3e0f a0 00    loop        ldy  #0
60:   3e11 a2 01                ldx  #1
60:   3e13 b1 03                lda  (3),y
60:   3e15 dd 42 03             cmp  834,x  ;stimmt erstes zeichen
60:   3e18 f0 16                beq  found
70:   3e1a 18       cont2       clc
70:   3e1b e6 03                inc  3
70:   3e1d d0 02                bne  cont1
70:   3e1f e6 04                inc  4
75:   3e21 a5 03    cont1       lda  3
75:   3e23 cd 41 03             cmp  833    ;schon am ende
75:   3e26 d0 e7                bne  loop   ;nein!
75:   3e28 a5 04                lda  4
75:   3e2a cd 42 03             cmp  834
75:   3e2d d0 e0                bne  loop   ;nein!
80:   3e2f 60                   rts         ;tschuess
90:   3e30 ec 3f 03 found       cpx  831    ;noch mehr zeichen
90:   3e33 f0 0c                beq  right
90:   3e35 e8                   inx
90:   3e36 c8                   iny
91:   3e37 b1 03                lda  (3),y
91:   3e39 dd 42 03             cmp  834,x  ;stimmt das naechste zeichen
91:   3e3c f0 f2                beq  found  ;ja
91:   3e3e 4c 1a 3e             jmp  cont2  ;nein
100:  3e41 18       right       clc
100:  3e42 a5 03                lda  3
100:  3e44 69 01                adc  #1
100:  3e46 8d 3c 03             sta  828
100:  3e49 a5 04                lda  4
100:  3e4b 69 00                adc  #0
100:  3e4d 8d 3d 03             sta  829
100:  3e50 a9 ff                lda  #255
100:  3e52 8d 3e 03             sta  830
100:  3e55 60                   rts         ;bye
U3e00-3e56
no errors

ready.
```

Assembler-Listing GET

```
profi-ass 64 v2.0    seite 1

20:   3c00                      .opt p,oo
20:   3c00                      *=   $3c00
30:   ffc6          chkin       =    $ffc6
40:   ffcf          basin       =    $ffcf
50:   ffcc          clrch       =    $ffcc
60:   8500          anfang      =    $8500
100:  3c00 a2 02    get         ldx  #2
100:  3c02 20 c6 ff             jsr  chkin    ;eingaberoutine auf floppy legen
110:  3c05 a0 00                ldy  #0
120:  3c07 20 cf ff loop        jsr  basin    ;zeichen von floppy holen
120:  3c0a 99 00 85             sta  anfang,y
120:  3c0d c8                   iny
120:  3c0e c0 ff                cpy  #255
120:  3c10 d0 f5                bne  loop
130:  3c12 20 cc ff             jsr  clrch    ;eingabe wieder normal setzen
130:  3c15 60                   rts           ;ciau
U3c00-3c16
no errors

ready.
```

Fortsetzung Seite 22

3000 Mark wa



Hier r
Sie k

und na

Wichtig: Werfen Sie einen
damit Ihr Programm auch

De



rten auf den Gewinner!

Der Wettbewerb geht weiter.

noch einmal kurz die Bedingungen:
önnen einsenden:

- Grafikprogramme
- Musikprogramme
- Spiele-
- Lernprogramme
- Anwenderprogramme

türlich völlig neue Programmideen.

Blick in das Kapitel "Hinweise für Autoren",
nerhalb von INPUT 64 lauffähig ist.

r Rechtsweg ist wie immer ausgeschlossen.

Assembler-Listing HEX

```
profi-ass 64 v2.0    seite 1

20:    3d00                      .opt p,oo
30:    3d00                 *=   $3d00
40:    ffd2         bsout   =    $ffd2
100:   3d00 20 67 3d        jsr  return
100:   3d03 ad 84 03        lda  900      ;anfangsadresse einlesen
100:   3d06 85 03           sta  3
100:   3d08 ad 85 03        lda  901
100:   3d0b 85 04           sta  4
100:   3d0d a2 00           ldx  #0
110:   3d0f a0 00   loop    ldy  #0
110:   3d11 20 6d 3d        jsr  prompt
110:   3d14 8a              txa
110:   3d15 20 85 3d        jsr  hex      ;ausgabe position
110:   3d18 20 61 3d        jsr  space
110:   3d1b 20 61 3d        jsr  space
120:   3d1e b1 03   loop1   lda  (3),y
120:   3d20 20 85 3d        jsr  hex      ;hex-ausgabe
120:   3d23 20 61 3d        jsr  space
120:   3d26 c8              iny
120:   3d27 c0 08           cpy  #8
120:   3d29 d0 f3           bne  loop1
130:   3d2b a0 00           ldy  #0
140:   3d2d b1 03   loop2   lda  (3),y
140:   3d2f 20 a5 3d        jsr  ascii    ;ascii-ausgabe
140:   3d32 c8              iny
140:   3d33 c0 08           cpy  #8
140:   3d35 d0 f6           bne  loop2
150:   3d37 20 67 3d        jsr  return
150:   3d3a 18              clc
150:   3d3b a5 03           lda  3
150:   3d3d 69 08           adc  #8
150:   3d3f 85 03           sta  3
150:   3d41 a5 04           lda  4
150:   3d43 69 00           adc  #0
150:   3d45 85 04           sta  4
150:   3d47 18              clc
150:   3d48 8a              txa
150:   3d49 69 08           adc  #8
150:   3d4b aa              tax
155:   3d4c a5 c5   stop    lda  197      ;tastaturabfrage
155:   3d4e c9 40           cmp  #64
155:   3d50 d0 fa           bne  stop
160:   3d52 a5 04           lda  4        ;am endeprint
160:   3d54 cd 87 03        cmp  903
160:   3d57 90 b6           bcc  loop     ;nein!
160:   3d59 a5 03           lda  3
160:   3d5b cd 86 03        cmp  902
160:   3d5e 90 af           bcc  loop     ;nein!
160:   3d60 60              rts
1000:               ;
1001:               ; *** versch. zeichenaugaben
1001:  3d61
1002:               ;
1020:  3d61 a9 20   space   lda  #32
1020:  3d63 20 d2 ff        jsr  bsout
1020:  3d66 60              rts           ; >space<
1040:  3d67 a9 0d   return  lda  #13
1040:  3d69 20 d2 ff        jsr  bsout
```

```
profi-ass 64 v2.0    seite 2

1040:  3d6c 60              rts           ; >return<
1060:  3d6d a9 3e   prompt  lda  #62
1060:  3d6f 20 d2 ff        jsr  bsout
1060:  3d72 60              rts           ; >promptzeichen
1080:  3d73 a9 12   rvson   lda  #18
1080:  3d75 20 d2 ff        jsr  bsout
1080:  3d78 60              rts           ; >rvs on<
1100:  3d79 a9 92   rvsoff  lda  #146
1100:  3d7b 20 d2 ff        jsr  bsout
1100:  3d7e 60              rts           ; >rvs off<
1102:  3d7f a9 2e   frage   lda  #46
1102:  3d81 20 d2 ff        jsr  bsout
1102:  3d84 60              rts           ; >'punkt'<
1109:               ;
1110:               ; * ausgabe einer hexzahl
1111:               ;
1120:  3d85 48      hex     pha
1120:  3d86 4a              lsr
1120:  3d87 4a              lsr
1120:  3d88 4a              lsr
1120:  3d89 4a              lsr
1120:  3d8a 20 94 3d        jsr  numout
1120:  3d8d 68              pla
1120:  3d8e 29 0f           and  #$0f
1120:  3d90 20 94 3d        jsr  numout
1120:  3d93 60              rts
1130:  3d94 c9 0a   numout  cmp  #$0a
1130:  3d96 b0 06           bcs  hexnum
1130:  3d98 09 30           ora  #$30
1130:  3d9a 20 d2 ff        jsr  bsout
1130:  3d9d 60              rts
1140:  3d9e 18      hexnum  clc
1140:  3d9f 69 37           adc  #$37
1140:  3da1 20 d2 ff        jsr  bsout
1140:  3da4 60              rts
1149:               ;
1150:               ; * ausgabe eines ascii zeichens
1151:               ;
1160:  3da5 48      ascii   pha
1160:  3da6 20 73 3d        jsr  rvson
1160:  3da9 68              pla
1160:  3daa c9 21           cmp  #33
1160:  3dac b0 07           bcs  cont1
1160:  3dae 20 7f 3d        jsr  frage
1160:  3db1 20 79 3d        jsr  rvsoff
1160:  3db4 60              rts
1170:  3db5 c9 80   cont1   cmp  #128
1170:  3db7 90 07           bcc  ausg
1170:  3db9 20 7f 3d        jsr  frage    ;uninteressante zeichen
1170:  3dbc 20 79 3d        jsr  rvsoff
1170:  3dbf 60              rts
1180:  3dc0 c9 22   ausg    cmp  #34
1180:  3dc2 f0 07           beq  hoch
1180:  3dc4 20 d2 ff cont3  jsr  bsout    ;ausgabe ascii
1180:  3dc7 20 79 3d        jsr  rvsoff
1180:  3dca 60              rts
1181:  3dcb a9 27   hoch    lda  #39
1181:  3dcd 4c c4 3d        jmp  cont3    ;anstatt des doppelhochkommas -> einf
achhochkomma
U3d00-3dd0
no errors

ready.
```

Rätselecke

Auflösung des Rätsels aus Ausgabe 7/85

Zugegeben, unser letztes Rätsel war eine ziemlich harte Nuß. Zwei ganze Lösungsvorschläge erreichten uns, eine davon leider falsch. Eindeutiger Gewinner des INPUT 64 Jahresabonnements ist:

Horst Fleckenstein aus Schiltach

Herzlichen Glückwunsch! Bei der eingeschickten Lösung handelte es sich um ein BASIC-Programm, das ein weiteres BASIC-Programm erzeugt, das direkt auf Diskette geschrieben wird. Mit diesem Programmgenerator lassen sich Lösungsprogramme für beliebige Kryptogramme der Form: SEND + MORE = MONEY erzeugen. Um allen INPUT 64-Lesern die Möglichkeit zu geben, den Programmgenerator einzusetzen, haben wir uns erlaubt, kleine Änderungen an dem Programm vorzunehmen. Nun wird das Lösungsprogramm direkt im Speicher ab Zeilennummer 10000 generiert. Sie können wählen, ob das erzeugte Programm vor der Ausführung erst abgespeichert werden soll oder nicht. Sie können auf Cassette, SuperTape (vorher einladen und initialisieren) oder auf Diskette speichern. Nach dem Abspeichern können Sie dann entscheiden, ob Sie weitere Lösungsprogramme erstellen oder das zuletzt erzeugte ausgeführen lassen wollen. Das Programm ermittelt alle richtigen Kombinationen. Gibt es keine Lösung, meldet es sich nach einiger Zeit ganz lapidar mit READY zurück.

Sie können das Programm wie gewohnt mit CTRL & S aus dem Magazin auf Ihren eigenen Datenträger abspeichern. Und dann flux an's Ausprobieren - spielt wirklich prima!

Und nun noch ein paar Erläuterungen zum unten abgedruckten Listing. Die Zeilen, die nur einen Doppelpunkt enthalten, sowie die Doppelpunkte am Zeilenanfang dienen nur zur Strukturierung des Programms. Möchten Sie den Programmablauf beschleunigen, können Sie diese entfernen. Zeile 99 rettet den Zeiger auf das Programmende des generierenden Programms. Zeile 100 setzt den Variablenanfangszeiger hoch, damit Platz für das zu generierende Programm freigehalten wird. Zeile 105 berechnet Hilfszeiger für das neue Programm und zum 'SAVEN' des Lösungsteils. Die Zeilen 110-340 erklären sich selbst. In den Zeilen 350-460 wird eine Plausibilitätsprüfung des eingegebenen Kryptogramms vorgenommen. Logische Fehler führen zum Neustart des Programms. Das neue BASIC-Programm wird in den Zeilen 490-1950 erzeugt. Das Programm wird in das String-Array A$(ZA) geschrieben, es enthält allerdings noch keine Zeilennummern. Zur Ablage von BASIC-Programmen im Speicher muß man wissen, daß beispielsweise der BASIC-Befehl THEN nicht durch die vier Zeichen T,H,E und N im Speicher dargestellt wird, sondern nur durch ein Zeichen. Diese Umwandlung der BASIC-Wörter nennt man 'Tokenisierung', das einzelne Zeichen dementsprechend TOKEN. So wird aus THEN eine 167, aus IF die Zahl 139 und so weiter. Tokenisiert wird normalerweise bereits bei der Eingabe der Programmzeilen, hier schreibt das Programm die TOKEN zunächst in das String-Array und später direkt in den Speicher. Beim LIST-Befehl geschieht übrigens genau das Gegenteil: die TOKEN werden in Klartext zurückgewandelt und so auf dem Bildschirm ausgegeben. Mit Hilfe der TOKEN-Tabelle und ein wenig Überlegung werden Sie jetzt entschlüsseln können, wie das Generatorprogramm die Kryptogramm-Lösung erzeugt.

Die Zeilen 1990-2230 dienen dazu, aus dem 'String-Programm' ein fast lauffähiges Programm zu machen. Hier werden die Zeilennummern errechnet und zugefügt. Gleichzeitig wird das Programm direkt hinter das Generator-Programm in den Speicher 'gepokt'. Ist es nun endlich fertig und kann anfangen zu rechnen? Nein, noch nicht! Aber gleich: Zeile 2250 ruft eine Kernal-Routine auf, die die BASIC-Zeilen 'bindet', das heißt, die BASIC-Zeilen werden richtig miteinander verknüpft. Jetzt kann das Lösungsprogramm mit GOTO 10000 angesprungen werden, und es arbeitet grad' so, als hätte es schon immer dort gestanden. Vor dem Start wird es noch abgespeichert, falls dies im Eingangs-Menue gewünscht wurde (Zeilen 2280 bis 2330). Es werden nur die neu erzeugten Zeilen ab 10000 'gesaved', diese können auch getrennt in den Rechner geladen und durch RUN gestartet werden. Sollen keine weiteren Programme generiert werden – die Abfrage steht in den Zeilen 2360-2370 – wird das zuletzt erstellte Programm gestartet. Nun etwas Geduld, und alle Lösungsmöglichkeiten werden Ihnen auf den Bildschirm ausgegeben.

Wahrscheinlich werden Sie jetzt fragen, warum dieser ganze Aufwand überhaupt getrieben wird. Der angewendete Trick besteht darin, daß die Buchstaben des Kryptogramms so als Variablen-Namen benutzt werden können. Aufwendige und komplizierte Abfragen auf Gleichheit und ähnliches entfallen damit. Eine geschickte Anordnung der Programmschleifen sorgt dann dafür, daß nicht sämtliche unmöglichen Kombinationen durchgespielt werden müssen. So erhält man in verblüffend kurzer Zeit alle Lösungen.

TOKEN – Tabelle

TOKEN	Befehl	TOKEN	Befehl
$80 128	END	$A6 166	SPC(
$81 129	FOR	$A7 167	THEN
$82 130	NEXT	$A8 168	NOT
$83 131	DATA	$A9 169	STEP
$84 132	INPUT#	$AA 170	+
$85 133	INPUT	$AB 171	-
$86 134	DIM	$AC 172	*
$87 135	READ	$AD 173	/
$88 136	LET	$AE 174	↑
$89 137	GOTO	$AF 175	AND
$8A 138	RUN	$B0 176	OR
$8B 139	IF	$B1 177	]
$8C 140	RESTORE	$B2 178	=
$8D 141	GOSUB	$B3 179	[
$8E 142	RETURN	$B4 180	SGN
$8F 143	REM	$B5 181	INT
$90 144	STOP	$B6 182	ABS
$91 145	ON	$B7 183	USR
$92 146	WAIT	$B8 184	FRE
$93 147	LOAD	$B9 185	POS
$94 148	SAVE	$BA 186	SQR
$95 149	VERIFY	$BB 187	RND
$96 150	DEF	$BC 188	LOG
$97 151	POKE	$BD 189	EXP
$98 152	PRINT	$BE 190	COS
$99 153	PRINT	$BF 191	SIN
$9A 154	CONT	$C0 192	TAN
$9B 155	LIST	$C1 193	ATN
$9C 156	CLR	$C2 194	PEEK
$9D 157	CMD	$C3 195	LEN
$9E 158	SYS	$C4 196	STR$
$9F 159	OPEN	$C5 197	VAL
$A0 160	CLOSE	$C6 198	ASC
$A1 161	GET	$C7 199	CHR$
$A2 162	NEW	$C8 200	LEFT$
$A3 163	TAB(	$C9 201	RIGHT$
$A4 164	TO	$CA 202	MID$
$A5 165	FN	$CB 203	GO

Neue Aufgabe:

Primen Sie 'mal!

Nachdem das vorige Rätsel nur die Informatik-Freaks angesprochen hat, diesmal wieder eine Aufgabe für den Normal-Programmierer . Es geht um das Zählen von Primzahlen. Alle Primzahlen? Und: was war das eigentlich nochmal, eine Primzahl? Sekunde, die 'Kleine Enzyklopädie Mathematik' sagt dazu:

Primzahlen sind Zahlen, die nur unechte Teiler haben; z.B. ist 5 nur durch 1 und 5, 13 nur durch 1 und 13 teilbar; 5 und 13 sind also Primzahlen. Die 1 selbst zählt man nicht zu den Primzahlen, so daß die Primzahlfolge mit 2 beginnt: 2,3,5,7,11,13,17,19,. . . (. . .) Unendlichkeit der Primzahlfolge. Schon EUKLID legte sich die Frage vor, ob die Folge von Primzahlen einmal abbricht oder ob es unendlich viele Primzahlen gibt. Er bewies indirekt, daß es keine größte Primzahl geben kann.

Was also eine Primzahl ist, haben wir damit geklärt, und alle sollen auch nicht gezählt werden, weil Sie sonst den Einsendeschluß am 5.12.85 (Datum des Poststempels; auch beim Brieftauben-Express) nicht einhalten könnten.

Die Aufgabe lautet also: Schreiben Sie ein BASIC-Programm, das die Anzahl der Primzahlen im Intervall von 0 bis 10000 ermittelt und diese sinnvoll auf dem Bildschirm ausgibt!

Natürlich können Sie auch wieder etwas gewinnen. Der Autor, der das schnellste Programm eingeschickt hat (bei gleicher Zeit entscheidet das Los), erhält ein JAHRESABONNEMENT INPUT 64. Außerdem warten diesmal 20 Buchpreise auf ihre Gewinner (10 neue Preise und 10 unvergebene Preise vom letzten Rätsel). Motiviert Sie das nicht, sich sofort auf die Aufgabe zu stürzen? Schicken Sie Ihre Lösung an die Redaktion (Anschrift im Impressum) und vergessen Sie nicht, Ihren Namen und die Laufzeit des Programms auf dem Listing anzugeben.

```
Programmgenerator

98 ifpeek(46)=80then100
99 poke2040,peek(45) : poke2041,peek(46)
100 printchr$(14)+chr$(8) : poke45,0 : poke46,80 : clr
105 up=peek(2041)*256+peek(2040)-2 : an=up : zt=10000
110 poke 53280,3 : poke 53281,3
120 print "          Programmgenerator"
130 print "          -------------------"
140 print"   zur Loesung eines Kryptogrammes"
150 print"      z.B.    S E N D"
160 print"              +  M O R E"
170 print"              -----------"
180 print"               M O N E Y"
190 print "Es wird ein Unterprogramm ab Zeile"
200 print"10000 generiert, das auch als eigen-"
210 print"staendig lauffaehiges Programm abge-"
220 print"speichert werden kann.         TASTE";
230 poke198,0:wait198,1:poke198,0
240 print"Auf welchen Datentraeger soll das"
250 print"Loesungsprogramm gespeichert werden?"
260 print"1 > Kassette":print"7 > SuperTape":print"8 > Diskette"
270 print"0 > Nicht speichern"
280 input"Geraeteadresse  8";ga
290 if ga=0 then310
300 input"Dateiname:";da$ : ifda$=""then ga=0
310 print""
320 input "  1.Summand:   send";s$(1)
330 input "  2.Summand:   more";s$(2)
340 input "  Summe:      money";s$(3)
350 for i = 1 to 3
360 :s(i)=len(s$(i))
370 next i
380 :
390 m1$=left$(s$(1),1)
400 m2$=left$(s$(2),1)
410 hi$=s$(1)+" + "+s$(2)+" = "+s$(3)
420 if s(1)>s(3) or s(2)>s(3) then run
430 if s(3)>s(1)+1ors(3)>s(2)+1 then run
440 if s(1)=s(3) or s(2)=s(3) then s(3)=s(3)+1 :s$(3)="0"+s$(3)
450 if s(1)<s(3) then s$(1)="0"+s$(1):s(1)=s(1)+1:goto450
460 if s(2)<s(3) then s$(2)="0"+s$(2):s(2)=s(2)+1:goto460
470 :
480 :
490 dim a$(200),b$(20)
500 za=0
510 a$(za)=chr$(153)+chr$(34)+"  Loesung des Kryptogrammes:"+chr$(34)
520 za=za+1
530 :
540 a$(za)=chr$(153)+chr$(34)+"  "+hi$+""+chr$(34)
550 za=za+1
560 :
570 :
580 bz=0 : b$(0)="0"
590 mm$=left$(s$(3),1)
600 if mm$="0"then 680
610 bz=bz+1 :b$(bz)=mm$
620 a$(za)=mm$+chr$(178)+"1"
630 za=za+1
640 :
650 :
660 :
670 :
680 for i = 1 to s(3)
690 :ne$=""
700 :a$(za)=chr$(0)
```

Fortsetzung Seite 26

Fortsetzung von Seite 25

```
710 :for l = 1 to 3
720 : t$(l)=mid$(s$(l),s(3)-i+1,1)
730 : fl=0 : rf=0
740 : for m = 0 to bz
750 :  if b$(m)=t$(l) then fl=1
760 : next m
770 : if fl<>0then ne$=" "+ne$:goto 1200
780 :
790 : bz=bz+1:b$(bz)=t$(l)
800 :
810 : if l=3 then rf=1 : goto 940
820 :
830 : su$="0"
840 : if t$(l)=m1$ or t$(l)=m2$ then su$="1"
850 :
860 : a$(za)=a$(za)+chr$(129)+t$(l)+chr$(178)+su$
870 : a$(za)=a$(za)+chr$(164)+"9"
880 : za=za+1
890 : ne$=t$(l)+ne$
900 : goto 1090
910 :
920 :
930 :
940 : a$(za)=a$(za)+t$(l)+chr$(178)+t$(l)+chr$(170)+t$(2)
950 : a$(za)=a$(za)+chr$(170)+"u("+right$(str$(i-1),1)+")"
960 : za=za+1
970 : ne$=" "+ne$
980 :
990 :a$(za)=chr$(139)+" "+t$(3)+chr$(179)+"10"
1000 :a$(za)=a$(za)+chr$(167)+"u("+right$(str$(i),1)+")"+chr$(178)+"0"
1010 :za=za+1
1020 :
1030 :a$(za)=chr$(139)+t$(3)+chr$(177)+"9"
1040 :a$(za)=a$(za)+chr$(167)+"u("+right$(str$(i),1)+")"+chr$(178)+"1:"+t$(3)
1050 :a$(za)=a$(za)+chr$(178)+t$(3)+chr$(171)+"10"
1060 :za=za+1
1070 :
1080 :
1090 : if bz=1 then 1200
1100 :
1110 : a$(za)=chr$(139)
1120 : for m = 1 to bz-1
1130 :  a$(za)=a$(za)+t$(l)+chr$(178)+b$(m)
1140 :  if m<>bz-1 then a$(za)=a$(za)+chr$(176)
1150 : next m
1160 :
1170 : a$(za)=a$(za)+chr$(167)+str$(i*200+100+(3-l)*10+zt)
1180 : za=za+1
1190 :
1200 :next l
1210 :
1220 :
1230 :
1240 :
1250 :if i=s(3) then 1530
1260 :
1830 :for l = 1 to 3
1840 : h$=mid$(ne$,l,1)
1850 : if h$=" "thena$(za)=a$(za)+":":goto1870
1860 : a$(za)=a$(za)+chr$(130)+h$
1870 : za=za+1
1880 :next l
1890 :
1900 :if i=1 then a$(za)=chr$(128)
1910 :if i<>1 then a$(za)=chr$(137)+str$(i*200-100+zt)
1920 :za=za+1
1930 :
1940 :
1950 next i
1960 :
1970 :
1980 :
1990 c=0:n=0:ln=0
2000 for i = 0 to za-1
2010 :h$=left$(a$(i),1)
2020 :if h$<>chr$(0)andh$<>chr$(1)then2070
2030 :a$(i)=right$(a$(i),len(a$(i))-1)
2040 :if h$=chr$(0) then c=c+1:n=0:ln=0
2050 :if h$=chr$(1) then n=n+1:ln=0
2060 :
2070 :zn=c*200+n*100+ln+zt
2080 :zh=int(zn/256):zl=zn and 255
2090 :
2100 :poke up,1 : poke up+1,8 :poke up+2,zl : poke up+3,zh : up=up+4
2110 : for j=1 to len(a$(i))
2120 :  poke up,asc(mid$(a$(i),j,1))
2130 :  up=up+1
2140 : next j
2150 :poke up,0 : up=up+1
2170 :ln=ln+10
2180 :
2200 :
2210 next i
2220 :
2230 poke up,0 : poke up+1,0
2240 :
2250 sys 42291
2260 :
2270 :
2280 if ga=0 then goto 10000
2290 :
2300 sys 57812 da$,ga
2310 ue=(up+2)/256 : poke 175,int(ue) : poke 174,(ue-int(ue))*256
2320 ue=int(an/256) : poke 194,ue : poke 193,(an-ue*256)
2330 sys 62954
2340 :
2350 :
2360 print"SQQQQQQNoch ein Programm erstellen  j]]]";
2370 input w$ : if w$="j"then run
2380 :

ready.
```

```
1270 :if rf=1 then 1450
1280 :
1290 :a$(za)=a$(za)+chr$(139)+t$(1)+chr$(170)+t$(2)+chr$(170)+"u("
1300 :a$(za)=a$(za)+right$(str$(i-1),1)+")"+chr$(178)+t$(3)
1310 :a$(za)=a$(za)+chr$(167)+"u("+right$(str$(i),1)+")"+chr$(178)+"0:"
1320 :a$(za)=a$(za)+chr$(137)+str$((i+1)*200+zt)
1330 :za=za+1
1340 :
1350 :a$(za)=chr$(139)+" "+t$(1)+chr$(170)+t$(2)+chr$(170)+"u("
1360 :a$(za)=a$(za)+right$(str$(i-1),1)+")"+chr$(178)+t$(3)+chr$(170)+"10"
1370 :a$(za)=a$(za)+chr$(167)+"u("+right$(str$(i),1)+")"+chr$(178)+"1:"
1380 :a$(za)=a$(za)+chr$(137)+str$((i+1)*200+zt)
1390 :za=za+1
1400 :
1410 :goto 1820
1420 :
1430 :
1440 :
1450 :a$(za)=chr$(137)+str$((i+1)*200+zt)
1460 :za=za+1
1470 :
1480 :
1490 :goto 1820
1500 :
1510 :
1520 :
1530 :a$(za)=a$(za)+chr$(139)+"u("+right$(str$(i-1),1)+") "+chr$(179)+chr$(177)
1540 :a$(za)=a$(za)+t$(3)+chr$(167)+str$(i*200-100+zt)
1550 :za=za+1
1560 :
1570 :a$(za)=chr$(153)+":"+chr$(153):za=za+1
1580 :
1590 :for l = 1 to 3
1600 : a$(za)=chr$(153)
1610 : for m = 1 to s(3)
1620 :  h$=mid$(s$(l),m,1)
1630 :  if h$ ="0" then h$=chr$(34)+"   "+chr$(34)
1640 :  a$(za)=a$(za)+h$
1650 :  if m<>s(3) then a$(za)=a$(za)+";"
1660 : next m
1670 : za=za+1
1680 :
1690 : if l<>2 then 1780
1700 : a$(za)=chr$(153)+chr$(34)
1710 : for m = 1 to s(3)
1720 :  a$(za)=a$(za)+"---"
1730 : next m
1740 : a$(za)=a$(za)+chr$(34)
1750 : za=za+1
1760 :
1770 :
1780 :next l
1790 :
1800 :
1810 :
1820 :a$(za)=chr$(1)
```

Loesung SEND+MORE=MONEY

```
10000 print"sqq Loesung des Kryptogrammes:"
10010 print"qq send + more = moneyq"
10020 m=1
10200 ford=0to9
10210 ifd=mthen 10320
10220 fore=0to9
10230 ife=more=dthen 10310
10240 y=d+e+u(0)
10250 if y<10thenu(1)=0
10260 ify>9thenu(1)=1:y=y-10
10270 ify=mory=dory=ethen 10300
10280 goto 10400
10300 :
10310 nexte
10320 nextd
10330 end
10400 forn=0to9
10410 ifn=morn=dorn=eorn=ythen 10520
10420 forr=0to9
10430 ifr=morr=dorr=eorr=yorr=nthen 10510
10440 ifn+r+u(1)=ethenu(2)=0:goto 10600
10450 if n+r+u(1)=e+10thenu(2)=1:goto 10600
10500 :
10510 nextr
10520 nextn
10530 goto 10300
10600 foro=0to9
10610 ifo=moro=doro=eoro=yoro=noro=rthen 10710
10620 ife+o+u(2)=nthenu(3)=0:goto 10800
10630 if e+o+u(2)=n+10thenu(3)=1:goto 10800
10700 :
10710 nexto
10720 :
10730 goto 10500
10800 fors=1to9
10810 ifs=mors=dors=eors=yors=nors=rors=othen 10920
10820 ifs+m+u(3)=othenu(4)=0:goto 11000
10830 if s+m+u(3)=o+10thenu(4)=1:goto 11000
10900 :
10910 :
10920 nexts
10930 goto 10700
11000 ifu(4) <>mthen 10900
11010 print:print
11020 print"    ";s;e;n;d
11030 print"    ";m;o;r;e
11040 print"---------------"
11050 printm;o;n;e;y
11100 :
11110 :
11120 :
11130 goto 10900

ready.
```

ID-Werkstatt

Zur Eröffnung unserer IDeen-Werkstatt stellen wir Ihnen zwei Einsendungen vor:

1.) Einen Flugsimulator in HIRES
2.) Eine Mathematische Formelsammlung in BASIC

Der Flugsimulator wurde uns von Herrn Schulze-Marmeling zugesandt. Sie können auch schon mit der vorliegenden Form Flugversuche unternehmen, jedoch ist das Programm noch nicht fehlerfrei. Das Programm können Sie auf Ihren Datenträger abspeichern und sich das Listing ansehen. Denken Sie daran, daß das Programm nur unter HIRESSPEED (Ausgabe 4/85) läuft! Also vorher HIRESSPEED laden und anstarten. Sie können nach Herzenslust Änderungen an dem Programm vornehmen. Herr Schulze-Marmeling würde sich über Zusendungen von besonders gelungenen Verbesserungen freuen. Seine Adresse finden Sie im Listing selbst.

Die mathematische Formelsammlung von Thomas Langenkamp kann einmal als Rechenprogramm und Trainer verwendet werden, zum anderen können gewiefte Programmierer sich einzelne Formeln aus dem Programm herauslösen und für eigene hochmathematische Anwendungen verwenden. Dieses Programm ist ebenfalls abspeicherbar.

Zum Abspeichern der Programme wählen Sie das entsprechende Modul mit der CURSOR-Taste im Inhaltsverzeichnis der ID-Werkstatt an und betätigen CTRL + S.

Bitte denken Sie daran!

Werkstatt-Produkte werden von der Redaktion n i c h t betreut. Wir können auch keine Fragen zur Bedienung, zu Programmfehlern oder Ähnliches beantworten. Wenn die Adresse im Programm veröffentlicht ist, können Sie direkt mit dem Autor Kontakt aufnehmen.

HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise: Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:

Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter. Das Programm sollte in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden. Sie können allerdings auch eine andere Hochsprache (zum Beispiel PASCAL) verwenden, für die es möglich ist, selbständig lauffähige Compilate zu erzeugen. Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein. Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht. Senden Sie uns Ihre Programme bitte auf Kassette oder Diskette mit kommentiertem Listing und Beschreibung. Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlichtlich erschwert wurde, zukünftig ungeprüft zurücksenden. Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse: 1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0800-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF). 2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein. 3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein. 4. Die CTRL-Taste darf nicht benutzt werden. Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

Hinweise zur Bedienung

1.) Entfernen Sie – bei ausgeschaltetem Rechner – eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD INPUT* ,8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise (Bitte Band zurückspulen und so weiter . . .).

4.) Merken Sie sich fünf Kommandos:
– (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

– (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum Band zurückspulen aufgefordert. Von Diskette wird automatisch neu geladen.)

– (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

– (CTRL) und (s) steht für SAVEn und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwendungsprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisungen – Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programmanfang möglich – entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft. Außerdem ist dies, um der Anfertigung illegaler Mehrfachkopien vorzubeugen, nur einmal pro Programm möglich.

– (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

– (CTRL) und (b) ermöglicht einen Bildschirmausdruck – natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Voraussetzung ist Geräteadresse 4.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist.

Auf der Rückseite der Verpackung finden Sie das Inhaltsverzeichnis der Kassette/Diskette.

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel Bei Ladeproblemen!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner abstürzen. Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

Bei Ladeproblemen:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft. Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben. Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

Tonkopf selbst justieren

Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme/Wiedergabekopf justieren können. Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kasset-

tenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage. Legen Sie nun die zweite Seite von INPUT 64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung PRESS PLAY ON TAPE kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar – bei nicht total verstellter Spurlage – völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am besten markieren), und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenden Vermerk an den Verlag ein (Adresse siehe Impressum).

Besitzer der Ausgabe 6/85 können es sich einfacher machen. Das dort veröffentlichte Programm RECORDER-JUSTAGE macht die Einstellung des Daten-Recorders zum Kinderspiel.

Listing justage

```
800 fori=49199to49410:readd:ps=ps+d:pokei,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys12*16↑3+11*16+10
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

Am 18.Dezember an Ihrem Kiosk:

INPUT 64 AUSGABE 12/85

Wir bringen unter anderem:

Rhythm-Maker

Der 64er als Rhythmus-Maschine: Ein Programm, mit dem Sie Ihrem Taktgefühl freien Lauf lassen können. Percussion-Sounds und Rhythmus-Sequenzen können von Ihnen entwickelt und auf Knopfdruck wieder abgerufen werden. Daß Sie diese Sequenzen auch 'saven' und wieder laden können, versteht sich fast von selbst.

Memory-List

Der Überblick über 64 Kilo-Byte RAM in Ihrem Rechner. Kein Sprite und kein Zeichensatz bleibt Ihnen verborgen.

Hilfsprogramme

Vom Bildschirm auf den Drucker: Hardcopy-Routinen für den C64 mit automatischer Zuweisung der Druckeradresse und der Fähigkeit, Groß-/Kleinschrift zu unterscheiden.

und außerdem:

Es geht weiter mit Nico durch die Denkwelt der Physik; im SID-Kurs wird hoch-, tief- und breitgefiltert; 64er Tips; Spiele und Anwenderprogramme fehlen natürlich auch diesmal nicht.

---

c't Magazin für Computertechnik

Ausgabe 12/85 - jetzt an Ihrem Kiosk

ECB-Bus-Adapter für Schneider CPC erschließt neue Hardware-Möglichkeiten :. Einführung in Small Talk :. Netzwerkanalyse mit C64 und Apple :. Pascal-Programme 'ROM-fähig' :. Serien: Einsteigen in CP/M, Dr. Osborne Kit, des Schneiders Kern :. u.v.a.m.

elrad-Magazin für Elektronik

Ausgabe 12/85 ab 25.November am Kiosk

Bauanleitungen: Präzisions-Funktionsgenerator :. Stereo-Equalizer :. Symmetrier-Box :. Test: Bausätze aus der Meßtechnik :. Schaltungskochbuch :. u.v.a.m.
(Änderungen vorbehalten)

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 1
Tel.: (05 11) 53 52-0

Technische Anfragen
nur dienstags von 9-16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Reprotechnik Hannover

Druck:
Leunisman GmbH, Hannover
CW Niemeyer Hameln

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198,-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

ISSN 0177-3771

Titelidee **INPUT 64**
Titelfoto: Pictor-Bildagentur München

Titel – Grafik und – Musik:
Tim Pritlove
Farbian Rosenschein

hier abtrennen

INPUT 64-Abonnement — Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat

____

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.
○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift ____________________

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement — Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl — Konto-Nr.

Geldinstitut — Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 61 04 07

3000 Hannover 61